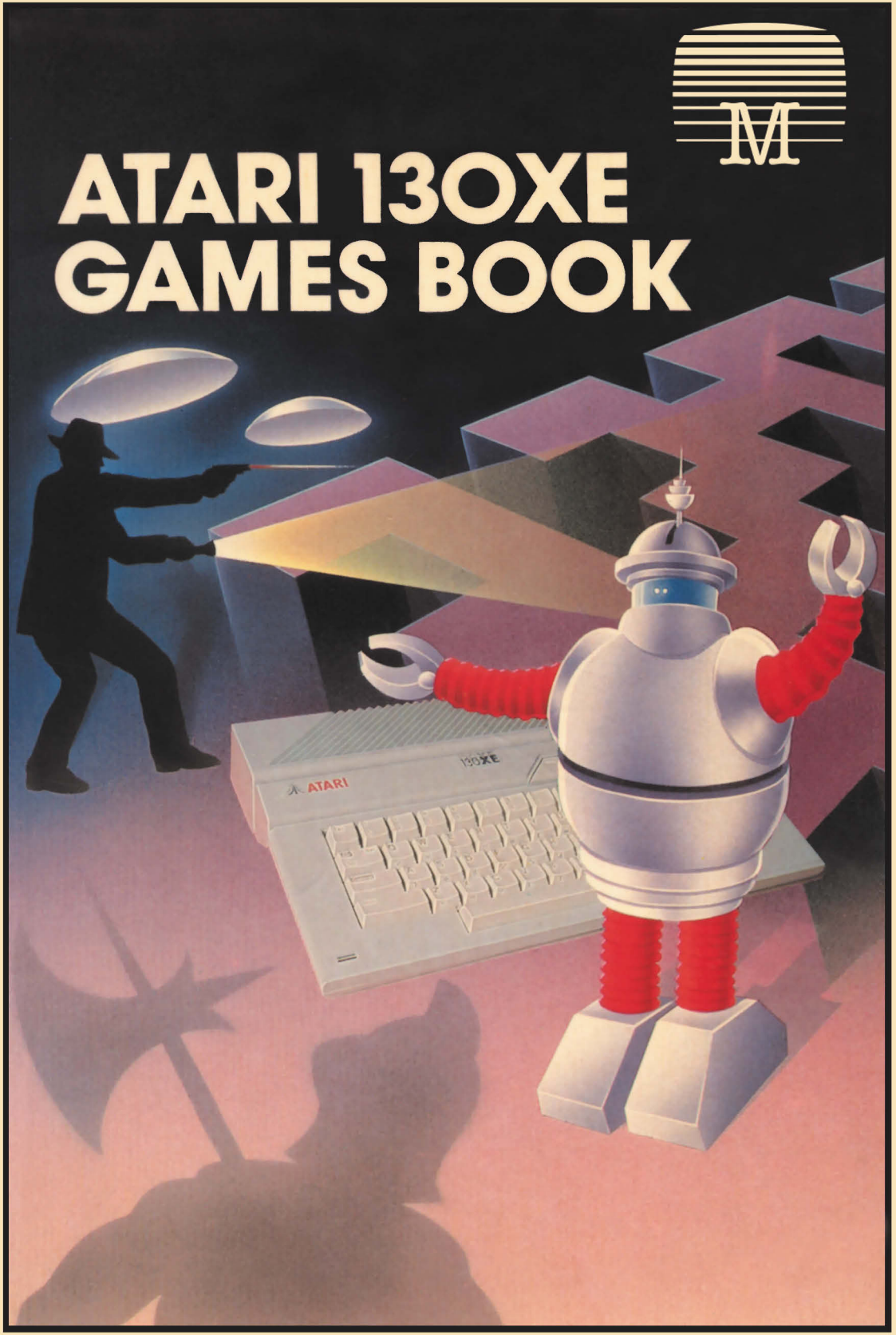
ATARI 130XE GAMES BOOK



CONTENTS

ATARI 130XE GAMES BOOK

ATARI 130XE GAMES BOOK

Richard Woolcock
&
Graeme Stretton

MELBOURNE HOUSE
PUBLISHERS

First Published in 1985 by Beam Software and Melbourne House

This Remastered Edition
Published by
Acorn Books
www.acornbooks.co.uk

This book is a page-by-page reproduction of the original 1985 edition as published by Beam Software and Melbourne House. The entirety of the book is presented with no changes, corrections nor updates to the original text, images and layout except for page 37 which has been reproduced in a similar style due to degradation of the original master; no guarantee is offered as to the accuracy of the information within.

ARRANGEMENT OF PROGRAMS

All the programs have been classified, explained and set out in an easy to read and enter format, with further programming suggestions and enhancements. We hope you enjoy this book and games within and continue to get the 'best' for and from your ATARI 13ØXE.

In the programs throughout this book, spaces have been used to aid readability. These have been placed between reserve words like PRINT, FOR, GOTO, GOSUB and between the characters in strings. It is not necessary to put them between reserve words most of the time however occasionally the machine will demand it. So if you type in a line omitting the spaces and the machine rejects it with a error, retype it with the spaces. The only time you should type a space inside of a string is when you see the * symbol. This avoids confusion.

The ATARI has a number of special graphics characters. These are obtained by pressing combinations of keys. The bulk of these characters are obtained by pressing the Control key and one of the alphabetic character keys. Inverse characters (reverse images of characters) are obtained by pressing the inverse key on the extreme bottom right hand side of the keyboard. Normal characters are restored by pressing this key once more.

Frequently occurring (and easily overlooked) typing errors with the ATARI 130XE

1. Do not confuse the letter O with the digit Ø (zero).

2. Do not confuse the capital letter I with the numeric digit 1 (one).

3. A comma and a full stop (period) are not interchangable.

4. When a colon is required do not type a semi-colon (;). These two characters are not interchangeable.

5. A double quote (") is not interchangeable with an apostrophe (').

6. Inside of character strings, spaces are mandatory if indicated by the * symbol.

7. It is important to get the number of brackets inside a BASIC formulae correct otherwise the line will be rejected. The bracket symbols are () and not [_.

8. The following characters are obtained by pressing the shift key and the numeric keys; ! " # $ % & ' @ ( )

Overall advice

If you type in a program line, press RETURN and the computer rejects it with an error message, then carefully compare the line with what's in the book. The line has been rejected because it has not been written according to the rules of BASIC. Retype the line correctly as per the book.

All BASIC program statements must be in upper case. Any reserve word in lower case will rejected as an error. Also reserve words may not be in the inverse mode.

Once you have typed in a program save a copy of it to tape or disk. Under no circumstances type in a program and RUN it without doing this first. Most of the programs in this book contain POKEs or machine language. If you make a mistake typing in a program and then RUN it, these are liable to erase your program or lock up the machine. If the error is disasterous enough, the only way to restart the machine is to switch it off and on, losing your program !!! If by some misfortune you should do this and the machine locks then press RESET. If control doesn't go back to BASIC then you have lost your program otherwise you may still have an oppotunity to save it to tape or disk.

Save a program to tape with

```
SAVE "C:FILENAME"
```

or to disk with

```
SAVE "D:FILENAME"
```

After you have typed in a program and saved it to either tape or disk, it's safe to RUN it. Unfortunately just because the computer has accepted a program line doesn't mean that it's correct. You are likely to be presented with a number of error messages the first time you try to RUN a program. To some extent this can be prevented by using CHEXSUM in the next section but even that won't solve all problems. Here is a list of the most common error messages and their probable causes.

ERROR- 17 AT LINE nnnn

This generally means that you have typed in a line, caused a syntax error and didn't notice it. When a syntax error occurs, the word ERROR- is entered into the start of the bad line. So when the ATARI tries to execute the line it finds garbage. The error is repaired by retyping in the line correctly.

ERROR- 12 AT LINE nnnn
The computer has been told to GOTO, GOSUB, ON GOSUB or ON GOTO to a line and the line didn't exist. Check that the line which has the above statements in it has the right linenumber. Then check that the line it was told to goto actually exists.

ERROR- 6 AT LINE nnnn
The computer tried to read some information from a DATA statement with a READ statement and there wasn't enough data present. The most obvious cause of this error is a mistake in the DATA statements. Carefully go through the DATA statements making sure that all numbers are right. Check to see that no full stops have been exchanged for commas and vice versa.

ERROR- 8 AT LINE nnnn
The computer tried to read information from DATA statements, was expecting numeric information and got character information instead. The solution to this problem is the same as above. Check through your data statements and make sure that all the information is correct. Also make sure that the READ statement where the error occured is correct.

ERROR- 3 AT LINE nnnn
The computer used a number which was out of range. For example a POKE statement tried to use a number which was not in the range Ø-255. If a POKE statement contains a variable then print the contents of the variable and find out how it got to that value. Generally happens when a READ statement fetches an incorrect DATA statement and the computer tries to POKE the bad data. Check the DATA statement.

ERROR- 9 AT LINE nnnn
A reference was made to an array or a string and an error occured. There are various reasons why this error has occured. They are:

* A reference was made to an array which didn't exist. There are two reasons for this; the variable in the line where the error occurred was incorrect, or the variable named in the DIM statement was incorrect. Check these two sources.

* An array reference was incorrect. It was either greater than 32767 or a negative number. Check that the array reference was in this range or was not greater than the dimension size.

* A string variable must be declared with a DIM statement at the start of the program. If you get an array error for a string then either the string variable where the error occurred is wrong or the varaible in the DIM statement is wrong. When you have typed in a program and you can't get it running properly, even after numerous debugging attempts, then put the job at rest for a day or so. It often happens that you will find the bug at once after resuming the job.

CHEXSUM

The unique CHEXSUM program validation

WHY

When a book of programs such as this book is keyed in, everybody invariably makes reading and typing mistakes and then spends ages trying to sort out where and what is causing the error (errors).

Even experienced programmers often cannot identify an error just by listing the relevant line and need to do the tedious job of going back to the book, especially with DATA statements.

Realizing that this is a major cause of frustration in keying the program, we decided to do something about it. There is a short routine in this book which you should key in and save BEFORE you key in any of the games programs.

Using this routine you will be able to find out if you made any keying errors at all and in which lines, before you even RUN the program. In effect this means that with this book you need not waste time looking for keying errors, you simply run the CHEXSUM routine and look at the display to identify lines containing errors. It's that easy.

The principle behind the routine is a unique check sum which is calculated on each line of the program you have keyed into the computer. Compare this chexsum value with the value for that line in the list at the end of the program listing; if they are the same the line is correct, if not there is an error in that line.

WHEN

The simplest method is to enter the CHEXSUM program in now and save a copy of it to tape or disk. To save it to disk use

```
LIST "D:CHEXSUM"
```

To save it to tape use

```
LIST "C:CHEXSUM"
```

The LIST command saves a copy of the CHEXSUM program to either tape or disk in ASCII. It is only possible to reload an ASCII file using:

For tape

```
ENTER "C:CHEXSUM"
```

For Disk

ENTER "D:CHEXSUM"

You can type in the CHEXSUM program at any time, even if you have started to type in a program. You cannot, of course LOAD in CHEXSUM from tape or disk because it will erase all you have typed so far. The obvious solution is to merge the programs. The CHEXSUM program should be saved onto a separate cassette to allow easy access.

HOW CAN YOU TELL IF CHEXSUM HAS BEEN ENTERED CORRECTLY

After having keyed in CHEXSUM it is very important that you know that CHEXSUM is working perfectly. Follow these instructions:

1. Type in the CHEXSUM program and save it to disk or tape with the commands suggested above.

2. Manually compare the CHEXSUM program you have typed in with the book. Get someone to read the book out to you while you check it against whats in the computer.

3. Keep repeating steps 1 and 2 until the checksum program is perfect.

Here is a listing of CHEXSUM and instructions on it's use:

```
32000 TOTAL=0
32010 STMTAB=PEEK(136)+PEEK(137)*256
32020 NUM=PEEK(STMTAB)+PEEK(STMTAB+1)*256
32030 IF NUM=32000 THEN GOTO 32070
32040 IF PEEK(STMTAB+4)=0 THEN 32050
32041 LINETOTAL=0:? "LINE ▲NUMBER: ▲▲";NUM;" ▲=▲";
32043 FOR T=STMTAB+4 TO STMTAB+PEEK(STMTAB+2)-1
32044 LINETOTAL=LINETOTAL+PEEK(T)
32045 NEXT T
32046 TOTAL=TOTAL+LINETOTAL
32049 ? LINETOTAL
32050 STMTAB=STMTAB+PEEK(STMTAB+2)
32060 GOTO 32020
32070 ? "TOTAL ▲=▲";TOTAL
```

USING CHEXSUM

CHEXSUM is a special program which generates a unique sum for each line in a program and a grand total of all sums. After each program listing is a table of checksums. You need only compare the numbers in the CHEXSUM table for each program with those generated by CHEXSUM. If two numbers differ, check that particular line.

1. Type in your game program, PINGPONG, say. Save it to tape or disk.

2. If you have just typed in a program then ignore this step otherwise LOAD in you game from tape or disk.

3. Merge the CHEXSUM program onto the end of your program. Do this by putting the tape or disk containing the chexsum program into the drive and for disk typing:

```
ENTER "D:CHEXSUM"
```

for tape type:

```
ENTER "C:CHEXSUM"
```

4. Once the CHEXSUM program has been merged onto the end of your game program, enter GOTO 32000 to activate CHEXSUM.

5. Chexsum will now output the checksum for the program. To halt the program press the Control and the '1' keys. Press again to restart output.

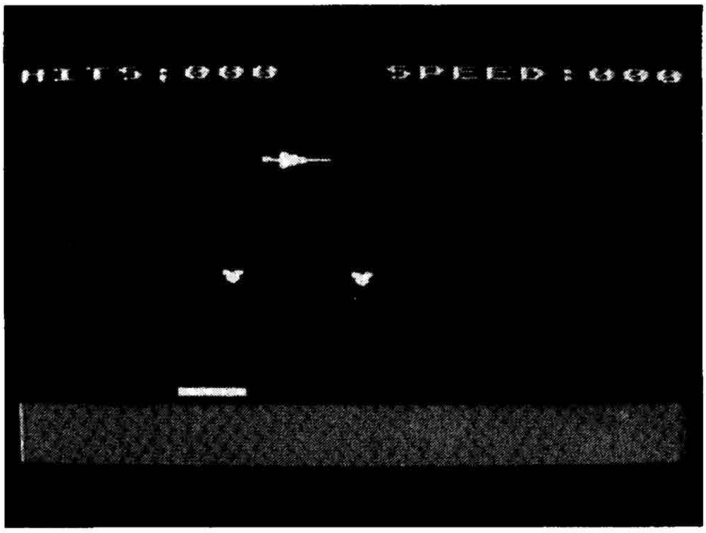
6. Check your grand total with that in the book. If they differ a line has been entered incorrectly. Compare line numbers until you locate the bad ones and then edit them.

7. Repeat steps 4 to 6 until the games program is debugged.

8. When the games program is running satisfactorily, delete the Chexsum program from the end of your game.

9. Finally save the debugged version onto a tape or disk.

BOMBER



CLASSIFICATION: Skill

A plane is flying above and periodically dropping bombs on the cities below. You have a shield which you must use to explode the bombs with before they hit the ground. The longer the game runs the faster the bomber flies and the faster the bombs are dropped. After a hundred bombs are dropped the speed decreases and after a hundred catches the speed increases. Use joystick one to move the shield left and right.

PROGRAMMING SUGGESTIONS

Have more than one bomber flying overhead and increase the number of bombs that can be dropped.

Program Variables

I	General purpose variable
PMBASE	Pointer to player missile data
PM	Page pointer to player missile data
A	Holds data begin read from data statement

Program Structure

5 - 8	Clear memory and read in programs
10 - 85	Set up graphics mode
10Ø - 12Ø	Data for players
1ØØØ	Call machine langugae program
5ØØØ - 541Ø	Data for machine language program

Listing

```
5       FOR I=33792 TO 33792+1023:POKE I,0:NEXT I
7       FOR I=30720 TO 30720+78:READ A:POKE I,A:NEXT I
8       FOR I=28672 TO 29510:READ A:POKE I,A:NEXT I
10      POKE 106,128
20      PM=PEEK(106):PMBASE=PM*256
30      GRAPHICS 1
35      PRINT #6;"CAUGHT: ▲▲▲▲HITS:"
40      POKE 559,62
50      POKE 53277,3
60      POKE 54279,PM
70      POKE 53256,2:POKE 53257,2
80      POKE 704,77:POKE 705,88:POKE 706,88
85      POKE 707,88
100     DATA 0,32,48,184,255,56,48,32,0,0,0,0,0,0,0,0,0,0,0,0
110     DATA 255,255,255,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
115     DATA 0,0,123,255,255,123,0,0,0,0,0,0,0,0,0,0,0,0,0,0
120     DATA 0,0,123,255,255,123,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1000    A=USR(112*256)
5000    DATA 32,27,112,32,72,112,32,151,112,32,41,113,32,239,11
        3,32,75,113,32,131
5010    DATA 113,32,206,113,76,3,112,169,0,141,5,115,169,50,141
        ,4,115,169,8,141
5020    DATA 7,115,169,50,141,19,115,169,150,141,18,115,169,0,1
        41,62,115,141,63,115
5030    DATA 141,66,115,141,56,115,169,5,141,65,115,96,173,60,1
        15,240,4,206,60,115
5040    DATA 96,173,65,115,141,60,115,173,7,115,201,4,240,5,201
        ,8,240,26,96,173
5050    DATA 5,115,240,4,206,5,115,96,169,8,141,7,115,32,127,11
        3,201,100,176,249
5060    DATA 141,4,115,96,173,5,115,201,163,240,4,238,5,115,96,
        169,4,141,7,115
5070    DATA 32,127,113,201,100,176,249,141,4,115,96,173,59,115
        ,240,4,206,59,115,96
5080    DATA 173,65,115,141,59,115,173,62,115,201,1,240,6,32,19
        7,112,76,182,112,32
5090    DATA 247,112,173,63,115,201,1,240,4,32,222,112,96,32,16
        ,113,96,32,127,113
5100    DATA 201,210,176,17,169,1,141,62,115,173,5,115,141,33,1
        15,173,4,115,141,32
5110    DATA 115,96,32,127,113,201,200,144,17,169,1,141,63,115,
        173,5,115,141,47,115
5120    DATA 173,4,115,141,46,115,96,173,32,115,201,155,240,4,2
        38,32,115,96,169,0
5130    DATA 141,62,115,169,220,141,32,115,238,56,115,96,173,46
        ,115,201,155,240,4,238
5140    DATA 46,115,96,169,0,141,63,115,169,220,141,46,115,238,
        56,115,96,32,247,114
5150    DATA 201,4,240,5,201,8,240,12,96,173,19,115,201,0,240,3
        ,206,19,115,96
5160    DATA 173,19,115,201,163,240,3,238,19,115,96,173,13,208,
        170,41,4,208,6,138
5170    DATA 41,8,208,20,96,169,0,141,62,115,169,220,141,32,115
        ,169,0,141,30,208
```

```
5180  DATA 238,66,115,96,169,0,141,63,115,169,220,141,46,115,
      169,0,141,30,208,238
5190  DATA 66,115,96,173,10,210,96,173,66,115,32,172,113,162,
      3,160,0,189,199,113
5200  DATA 153,135,125,200,202,208,246,173,56,115,32,172,113,
      162,3,160,0,189,199,113
5210  DATA 153,144,125,200,202,208,246,96,162,3,56,160,0,253,
      202,113,144,3,200,208
5220  DATA 248,125,202,113,72,152,9,16,157,199,113,104,202,20
      8,231,96,0,0,0,1
5230  DATA 10,100,173,66,115,201,100,240,8,173,56,115,201,100
      ,240,10,96,206,65,115
5240  DATA 169,0,141,66,115,96,238,65,115,169,0,141,56,115,96
      ,120,32,202,114,160
5250  DATA 14,162,0,189,253,114,149,176,232,136,208,247,32,10
      8,114,160,14,162,0,181
5260  DATA 176,157,253,114,232,136,208,247,160,14,162,0,189,1
      1,115,149,176,232,136,208
5270  DATA 247,32,108,114,160,14,162,0,181,176,157,11,115,232
      ,136,208,247,160,14,162
5280  DATA 0,189,25,115,149,176,232,136,208,247,32,108,114,16
      0,14,162,0,181,176,157
5290  DATA 25,115,232,136,208,247,160,14,162,0,189,39,115,149
      ,176,232,136,208,247,32
5300  DATA 108,114,160,14,162,0,181,176,157,39,115,232,136,20
      8,247,32,216,114,88,96
5310  DATA 165,183,197,182,240,68,160,0,165,184,24,105,46,145
      ,176,169,32,24,101,182
5320  DATA 168,166,185,169,0,145,178,200,202,16,250,169,32,24
      ,101,183,141,64,115,162
5330  DATA 0,142,53,115,166,185,172,53,115,177,180,238,53,115
      ,172,64,115,145,178,238
5340  DATA 64,115,202,16,237,165,183,133,182,165,184,133,189,
      96,165,184,197,189,208,182
5350  DATA 96,173,57,115,41,15,170,189,230,114,238,57,115,96,
      160,14,162,0,181,176
5360  DATA 157,68,115,232,136,208,247,96,160,14,162,0,189,68,
      115,149,176,232,136,208
5370  DATA 247,96,1,2,3,4,5,10,7,8,7,8,11,4,2,4,1,4,8,173
5380  DATA 0,211,73,255,96,0,208,0,132,0,120,0,0,0,8,0,16,0,0
      ,1
5390  DATA 208,0,133,20,120,0,0,0,8,0,16,0,0,2,208,0,134,40,1
      20,0
5400  DATA 0,0,8,0,16,0,0,3,208,0,135,60,120,0,0,0,8,0,0,0
5410  DATA 0,0,0,79,0,0,0,0,0,0,0,0,0,5,0,0,0,0,156
```

ChexSum Tables

```
   5 = 1421
   7 = 1494
   8 = 1568
  10 = 277
  20 = 1124
  30 = 144
  35 = 1218
  40 = 420
  50 = 406
  60 = 473
  70 = 782
  80 = 1203
  85 = 378
 100 = 2454
 110 = 2368
 115 = 2372
 120 = 2188
1000 = 716
5000 = 3467
5010 = 3393
5020 = 3601
5030 = 3556
5040 = 3405
5050 = 3450
5060 = 3358
5070 = 3611
5080 = 3549
5090 = 3484
5100 = 3533
5110 = 3590
5120 = 3507
5130 = 3706
5140 = 3623
5150 = 3285
5160 = 3509
5170 = 3446
5180 = 3616
5190 = 3567
5200 = 3748
5210 = 3630
5220 = 3390
5230 = 3619
5240 = 3559
5250 = 3703
5260 = 3866
5270 = 3745
5280 = 3714
5290 = 3769
5300 = 3719
5310 = 3726
5320 = 3761
5330 = 3776
5340 = 3956
5350 = 3625
5360 = 3788
5370 = 2398
5380 = 2652
5390 = 2736
5400 = 2390
5410 = 2037

TOTAL = 166569
```